Large-Scale Distributed Systems and Software Development and Productivity

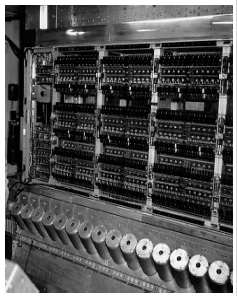
- **Rick Schantz, BBN (panel leader)**
- **David Sharp, Boeing**
- **Premkumar Devanbu, UCDavis**
- **Priya Narasimhan, CMU**
- **Mike Masters, NSWC**

Group Discussion and working group

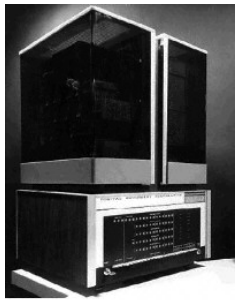
Questions

- What requirements/forces/changes/insights will be the drivers for moving forward or to new approaches?
- What is the current baseline for our capability to develop such systems (what can and can't we do)?
- What are the key problems/technologies and approaches to the problems/technologies which need to be addressed/overcome in moving forward?
- What would happen differently/be enabled if we overcame these problems?
- Where might we realistically expect to be in 2years, 5 years, 10 years, if we committed significant effort to these issues?

Historical Perspective:Software Infrastructure Enables Applications



Application

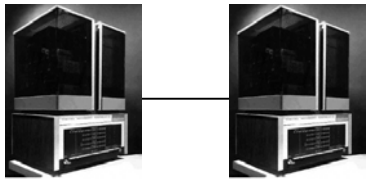


Application

Operating System

App App

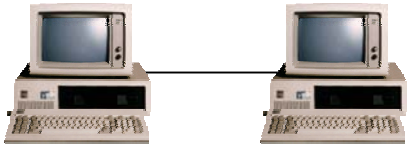
Operating System



App App

OS OS

Network Protocols

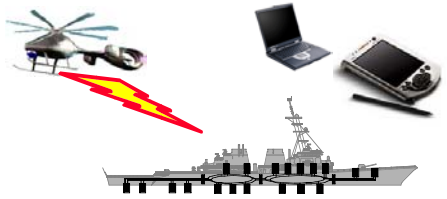


App App

Middle-ware MW Svcs MW Svcs Middle-ware

OS OS

Network Protocols



App App

Middle-ware MW Svcs MW Svcs Middle-ware

OS OS

Network Protocols

1950s **Fifty Years of Distributed Systems Software Architecture Evolution** 2001+

Forces Influencing Our Future:
Are We Prepared to go There?

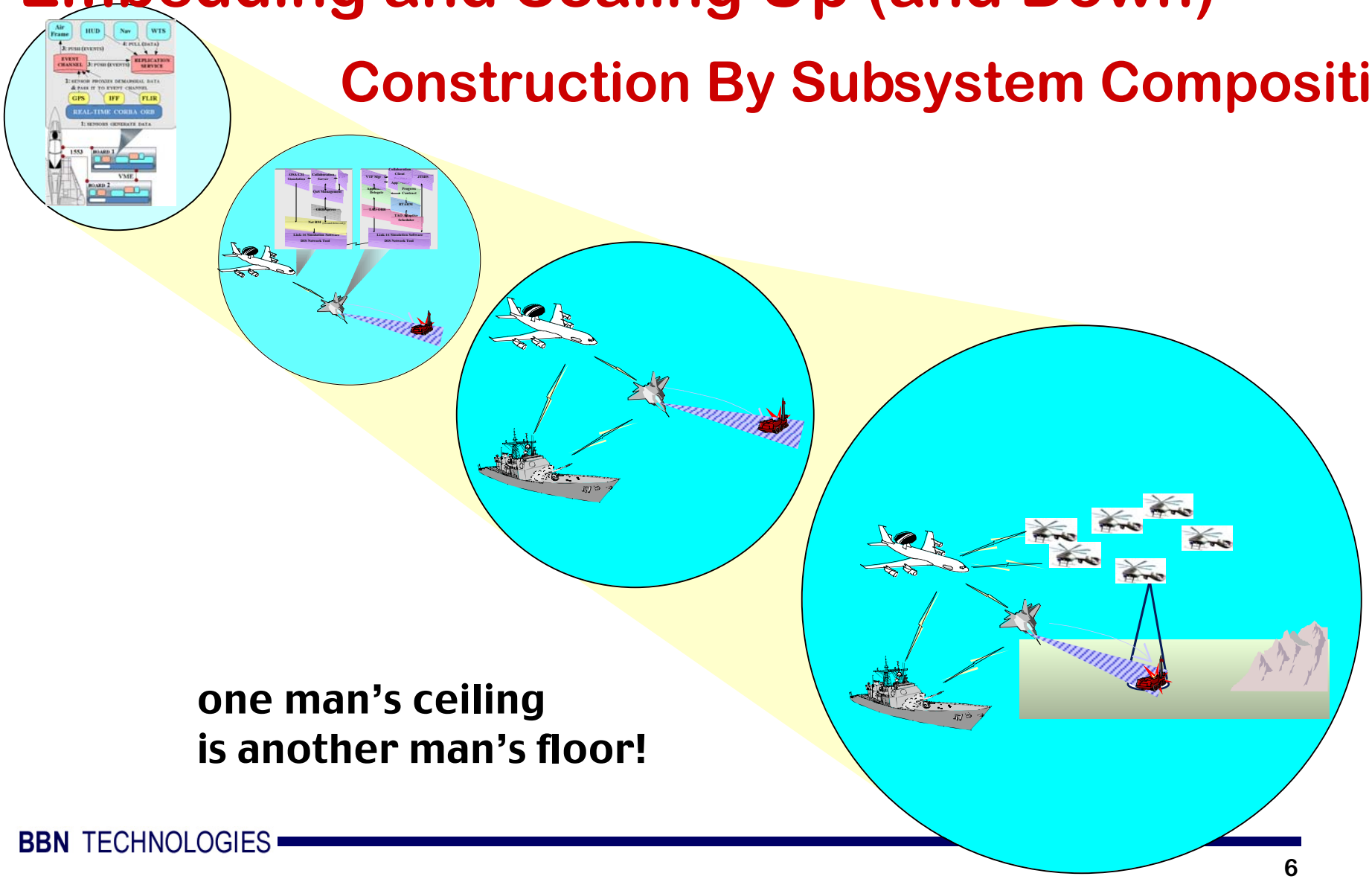
- Everything is a computer
- Everything is a networked computer
- Everything is potentially interdependent
- Things connect to the real world
- Increasing heterogeneity and scale
- Demand Driven Critical Systems will continue to be attempted with or without proper basis, understanding and tools

Some Example Applications

- Traffic control
  - Sensor data from 1000s of vehicles
- Theater battle management and dynamic replanning
- Supply chain management
- Automated vehicles and weapons
- Community analysis of scientific data
  - Soft-real-time response and query optimization from 1000s of users, via coordinated management of 1000s of resources
- Home power management
- Total Ship Computing
- …

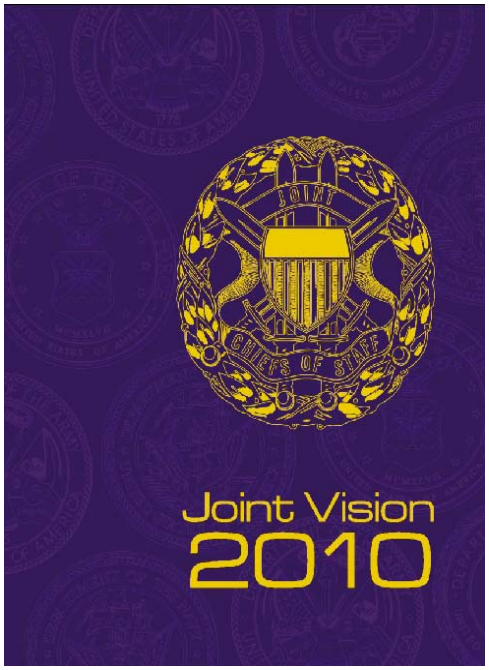
Embedding and Scaling Up (and Down)

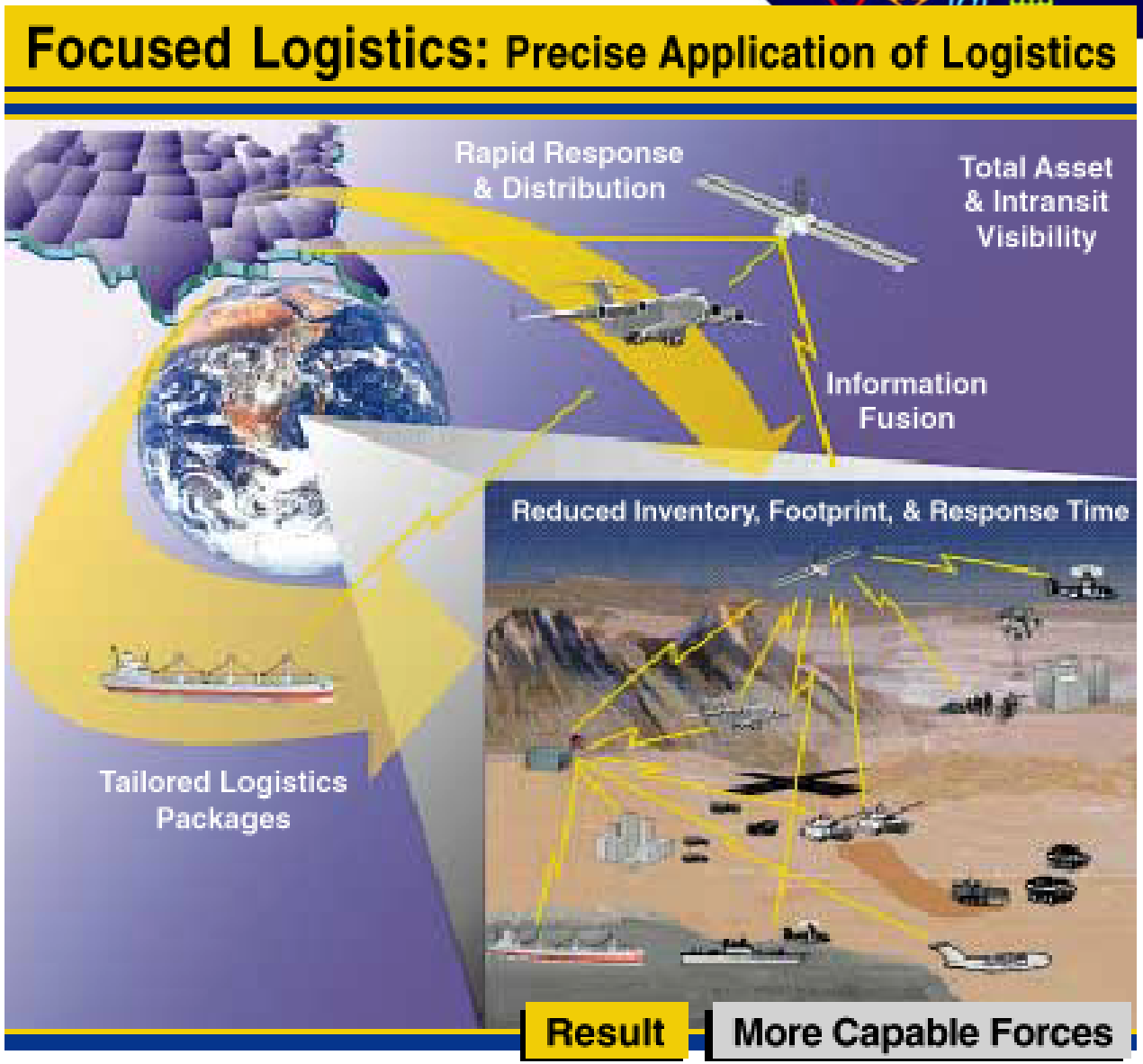
Construction By Subsystem Compositio



one man's ceiling
is another man's floor!

A Large Scale Information System Challenge

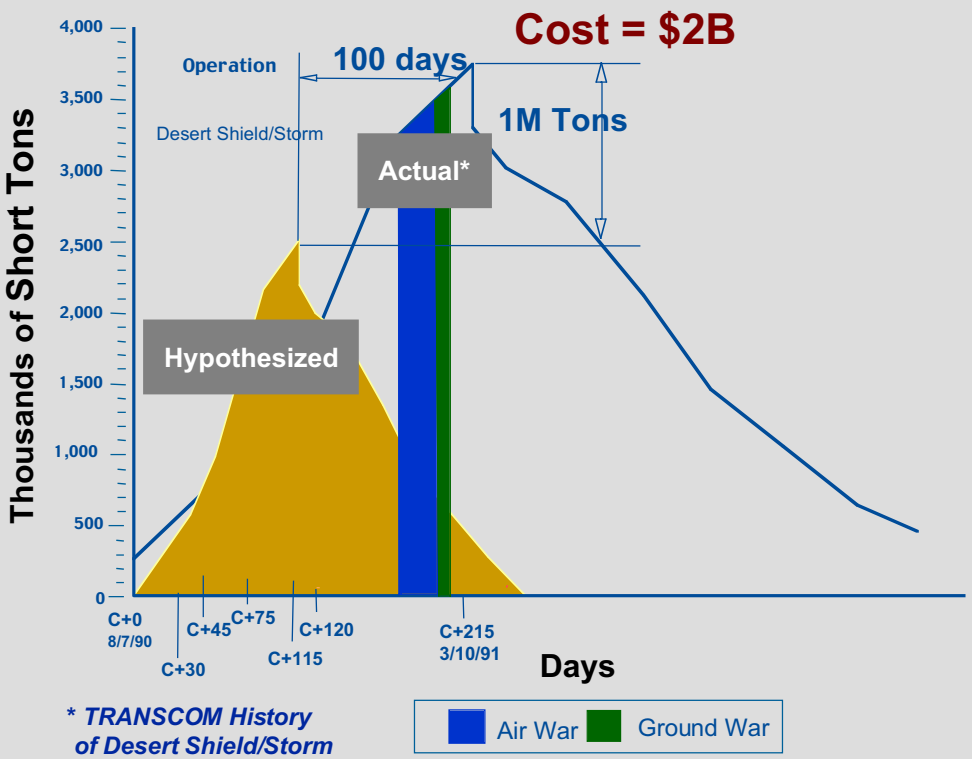




The Enormity of Military Logistics

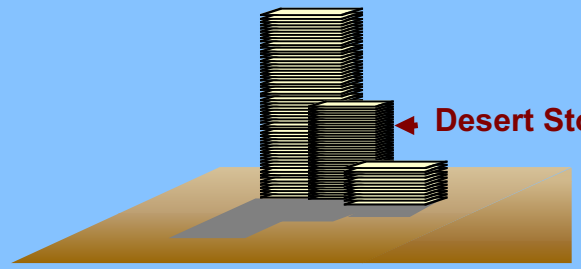
- **Military Logistics is Enormously…**
  - **Complex : 10000+ interacting/cooperating Organizations each with own business processes**
  - **Detailed : 6,000,000 NSN's (National Stock Numbers = Distinct Object Types)**
  - **Dynamic : Plans change as resources, requirements, execution change**

The Impact of Information Systems



* *TRANSCOM History of Desert Shield/Storm*

The Current Planning Environment



- Sequential phases
- Manually intensive
- Plans take days to months to complete
- Based on notional data
- Limited understanding of shortfalls and bottlenecks
- Plans are static artifacts…

Current systems take weeks to build low quality plans with notional data

…Plus ….

- ***No operational architecture***
- ***Over a 1,000 stovepipe, logistics systems***
- ***Vast majority of which are not interoperable, lack flexibility and are difficult to evolve to new processes and doctrine***

Taking Stock

- Connecting parts is easy, building end to end systems that work is hard
- Volatility in the computer environment as the rule, not the exception
- need for lots of customized network-centric applications using off-the-shelf building blocks to make them affordable, by regular system engineers (transition and availability of results are issues too!)
- interesting behavior is between the nodes -> emphasis on evolution of multi-level middleware in effecting a global behavior from coordinating and managing local behavior, and how it fits with basic programming concepts and environments
- trends
  - scaling way up (without reprogramming the pieces)
  - scaling way down (without reinventing the concepts)

Current Paths to Build On and New Directions

- develop a new computational model organized around dynamic, integrated, multiple property tradeoffs
- develop system composition techniques with policy controlled managed composite behavior to assure beneficial composition
- embedded support for validation, testing and safety over variable/adaptable conditions
- off-the-shelf distributed control capabilities
- defining appropriate scoping and granularity for integrated multi-layer resource management
- strategies for using throwaway low cost elements in large redundant numbers (1:many, many:many)
- variable trust models (who do you trust?)

"Able" Software Summary

Software Engineering breakthroughs and approaches to building systems which are

- Adaptable
- Scaleable
- Composable
- Certifiable
- Affordable

for large scale distributed environments

That's a Tall Order! Let's Get Going.